EDoF-ToF: extended depth of field time-of-flight imaging

JASPER TAN, VIVEK BOOMINATHAN, RICHARD BARANIUK, AND ASHOK VEERARAGHAVAN*

Electrical and Computer Engineering Department, Rice University, Houston, TX 77005, USA
**vashok@rice.edu*

Abstract: Conventional continuous-wave amplitude-modulated time-of-flight (CWAM ToF) cameras suffer from a fundamental trade-off between light throughput and depth of field (DoF): a larger lens aperture allows more light collection but suffers from significantly lower DoF. However, both high light throughput, which increases signal-to-noise ratio, and a wide DoF, which enlarges the system's applicable depth range, are valuable for CWAM ToF applications. In this work, we propose EDoF-ToF, an algorithmic method to extend the DoF of large-aperture CWAM ToF cameras by using a neural network to deblur objects outside of the lens's narrow focal region and thus produce an all-in-focus measurement. A key component of our work is the proposed large-aperture ToF training data simulator, which models the depth-dependent blurs and partial occlusions caused by such apertures. Contrary to conventional image deblurring where the blur model is typically linear, ToF depth maps are nonlinear functions of scene intensities, resulting in a nonlinear blur model that we also derive for our simulator. Unlike extended DoF for conventional photography where depth information needs to be encoded (or made depth-invariant) using additional hardware (phase masks, focal sweeping, etc.), ToF sensor measurements naturally encode depth information, allowing a completely software solution to extended DoF. We experimentally demonstrate EDoF-ToF increasing the DoF of a conventional ToF system by 3.6×, effectively achieving the DoF of a smaller lens aperture that allows 22.1× less light. Ultimately, EDoF-ToF enables CWAM ToF cameras to enjoy the benefits of both high light throughput and a wide DoF.

1. Introduction

3D measurements have been shown to be useful in various applications, ranging from entertainment to security to healthcare and others. In particular, the continuous-wave amplitude-modulated time-of-flight (CWAM ToF, referred to as *ToF* hereon) imaging system [1] has proven to be a popular method for acquiring such 3D measurements because of its compactness, low cost, and robustness. ToF cameras compute depth maps by emitting temporally coded illumination into the scene and measuring the illumination's time delay as it is reflected by the objects in the scene. In recent years, ToF systems have been shown to benefit various applications such as action recognition [2], robot navigation [3], respiratory motion tracking [4], smart room human tracking [5], product quality inspection [6], and face recognition [7].

Problem statement. Since ToF imaging systems rely on conventional optical lenses, they suffer from these lenses' inherent trade-off between depth of field (DoF) and signal-to-noise ratio (SNR), as shown in Fig. 1 (left). In particular, lenses with larger apertures (e.g. $f/1.7$) provide higher light throughput (and thus increased SNR) at the expense of a narrower DoF. Using a smaller aperture (e.g. $f/8$) widens the DoF but receives less light. As a result, as shown in Fig. 1 (middle), one must choose a lens that provides either a high SNR or a wide DoF, but not both.

The trade-off between DoF and SNR is especially harmful to ToF systems since they rely on active illumination. Because of the R^2 fall-off of active illumination, objects further away receive and reflect less photons, exacerbating the low light throughput of small-aperture lenses.

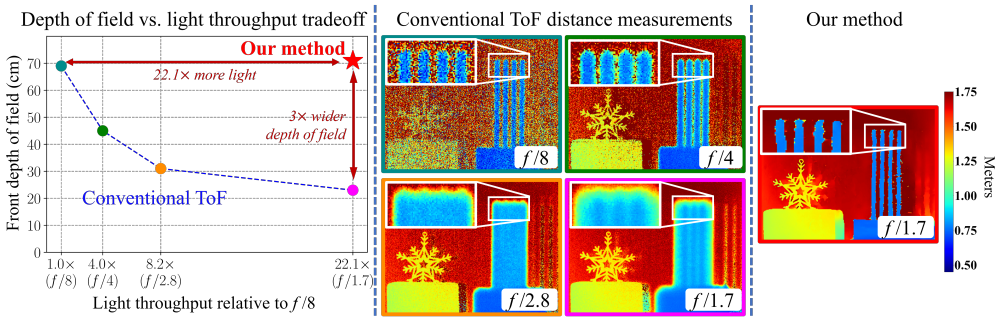


Fig. 1. Left: lenses with larger apertures (e.g. $f/1.7$) provide higher light throughput but at the expense of a narrow DoF (quantified by the experiment detailed in Section 5.3). Middle: sample conventional ToF depth maps captured with different lens apertures illustrate the fundamental SNR vs. DoF trade-off of conventional ToF systems. Right: EDoF-ToF provides depth maps that have both low noise and a wide DoF. All measurements shown are taken with a 35mm lens focused at 1.4m.

While one could address the low SNR by increasing illumination power or exposure time, such solutions introduce additional problems such as higher on-device power requirements or motion blur errors. Indeed, it is highly desirable to equip ToF systems with large-aperture lenses while addressing the resulting narrow DoF.

In this work, we propose *Extended Depth of Field Time-of-Flight (EDoF-ToF)*, an algorithmic method that enables ToF systems to enjoy both a wide DoF and high light throughput by pairing a conventional large-aperture lens with a deblurring neural network. The large-aperture lens enables high light throughput and high SNR, and the neural network extends its DoF by deblurring defocused regions in the ToF measurements. The key component to EDoF-ToF's success is a simulator that can accurately generate large-aperture ToF training data for the neural network. This simulator efficiently models the depth-dependent defocus blurring and partial occlusions that arise from a large aperture. A core benefit of EDoF-ToF is that it requires no optical or hardware modifications to the conventional ToF system, as opposed to methods typically employed for extending the DoF of conventional photography or microscopy systems [8–10].

We experimentally evaluate EDoF-ToF on a Texas Instruments OPT8241 ToF system with a 35mm $f/1.7$ lens and demonstrate a 3.6× increase in the DoF. Essentially, EDoF-ToF achieves the DoF of a conventional $f/8$ system while receiving 22.1× higher light throughput, thus producing low-noise all-in-focus depth maps, as shown in Fig. 1 (right). Further, we experimentally demonstrate that EDoF-ToF successfully inverts some of the largest blurs addressed in the ToF literature to date.

ToF signal-to-noise ratio vs. depth of field trade-off

The trade-off between SNR and DoF lies in the lens aperture size: a larger aperture yields a higher SNR by increasing light throughput but a narrower DoF. A detailed discussion of a ToF system's noise characteristics can be found in Section 3 of [11], where the authors show the SNR of the ToF system's phase measurement (and thus depth map) to be:

$$\text{Phase SNR} \propto \frac{(n_e - n_a)C_d}{\sqrt{n_e + n_T^2}},$$

where n_e is the number of generated electrons from all received photons, n_a is the number of electrons generated only from ambient illumination, C_d is the demodulation contrast between the

ToF pixel's nodes, and n_T is the pixel capacitor's reset noise. For a fixed wavelength, the number of electrons generated by a sensor pixel increases linearly with the number of photons hitting the active pixel area. Since the number of photons increases linearly with the aperture area, which increases linearly with the square of the aperture diameter, we have that $n_e, n_a \propto d^2$, where d is the lens aperture diameter. Thus, for sufficiently low reset noise ($n_T^2 << n_e$), the phase SNR increases linearly with d:

$$\text{Phase SNR} \propto C_d d.$$

The DoF is defined as the depth range at which the blur due to defocus is sufficiently small. One way to quantify DoF is by finding the depths where the circle of confusion is below some threshold C under geometrical approximations. By applying the geometry of similar triangles to the geometric image formation model, [12] derives an approximation of the DoF:

$$DoF \approx \frac{2u^2vC}{f^2d},$$

where u is the distance from the first principal plane to the focal plane, v is the distance from the second principal plane to the image sensor, and f is the focal length. Thus, the DoF is inversely proportional to the aperture diameter d.

Therefore, the phase SNR is proportional to the aperture diameter, but the DoF is inversely proportional to it. Therein lies the fundamental trade-off that EDoF-ToF addresses.

2. Related work

Extending DoF. Since the SNR vs. DoF trade-off is inherent in conventional optical lenses, it affects any imaging system that relies on such lenses such as conventional photography [10], stereo vision [13], and microscopy [8,9,14]. Generally, defocus can be addressed by deconvolving the measurement with the defocus blur point spread function (PSF). However, since a lens's defocus PSF varies with scene depth and the depth at each pixel is unknown, the blur PSF for each pixel is generally unknown. A popular idea for extending the DoF of imaging systems is to remove the depth-dependence of the defocus blur by incorporating hardware modifications such as focal sweeping [8, 10], depth-dependent chromatic aberrations [15], optical diffusers [16], or designed phase masks [9, 17]. Another idea is to learn phase masks and a reconstruction neural network in an end-to-end fashion to produce all-in-focus measurements [13, 14]. On the other hand, EDoF-ToF is a purely digital method that requires no hardware modifications. We hypothesize that this is made possible by the depth information contained in ToF measurements, alleviating some of the depth-dependent PSF ambiguity at each pixel.

Extending DoF for ToF. Other researchers have also proposed methods for extending the DoF of ToF systems by employing hardware modifications. In [18], the authors propose using additional hardware such as a microlens, a mask, angle-sensitive pixels, or a camera array to capture light fields, which allow synthetic refocusing. Alternatively, the work of [19] sweeps the focal plane of a standard ToF camera during a single capture to have a single blur PSF for all depths and then deconvolving this individual PSF using total-variation regularized non-blind deconvolution. In contrast, EDoF-ToF is a purely software method that requires no additional mechanics or hardware modifications (apart from using a large-aperture lens). As such, it has the benefit of not adding any bulk, weight, or hardware complexity to the camera and being readily applicable to any existing ToF system.

The work of [20] is similar to ours in that it extends the DoF of ToF systems with a purely software method. Its method consists of calibrating for the blur PSFs at different depths and then iterating through estimating each pixel's blur PSF and deconvolving these estimated blur PSFs through an optimization scheme based on the alternating direction method of multipliers

(ADMM). EDoF-ToF differs in that it takes advantage of the strength of deep learning techniques to solve inverse problems and can deblur larger blur kernels than shown in [20].

Deep Learning for ToF. Deep learning techniques have also been employed by other researchers to improve other aspects of ToF systems such as multi-path interference correction and denoising [21–26], confidence estimation [27], translucent object error correction [28], phase unwrapping [22], flying pixel correction [29], and ToF-RGB alignment [30]. To the best of our knowledge, our work is the first to use deep learning to extend the DoF of ToF cameras.

Image Deblurring. Deblurring conventional camera images has been well-explored in literature (see, for example, the survey of [31]). In recent years, deep learning has been applied to the task of image deblurring with great success [32–34]. In this work, we use the scale-recurrent network (SRN) architecture, a deep learning deblurring system introduced by [33] that achieves state-of-the-art image deblurring.

3. Simulating large-aperture ToF measurements

An efficient procedure of simulating accurate large-aperture ToF measurements can produce abundant training data, allowing one to leverage deep learning methods to extend the DoF of real ToF measurements. Two challenges need to be addressed to design such a simulator. First, the optical consequences of large apertures must be modeled in a computationally feasible manner. Second, these optical consequences must be incorporated into the ToF's nonlinear forward model.

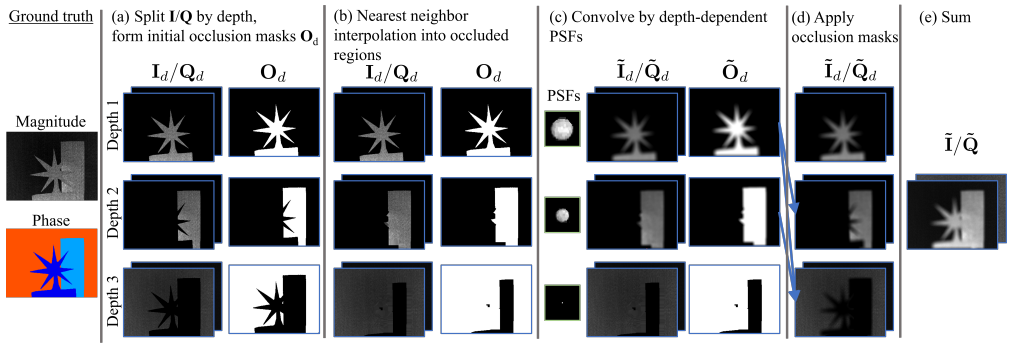
We address these two challenges with a number of techniques described in this section. First, we mathematically derive how defocus blur affects ToF measurements while leveraging the measurements' phasor representations to efficiently simulate this blurring. Next, we address partial occlusions by approximating an occlusion mask based on the blur point-spread functions (PSFs) and by using simple linear operations and nearest neighbor interpolation for computational efficiency. Finally, we estimate our lens's real defocus blur PSFs at different depths using a calibration procedure based on random noise patterns. Ultimately, the simulator consists of five steps, as illustrated in Fig. 2a: (a) splitting measurements by depth, (b) nearest neighbor interpolation of occluded regions, (c) convolution by depth-dependent blur PSFs, (d) elementwise multiplication of transparency masks, and (e) summation of the depth components.

3.1. Depth-dependent defocus blurring of ToF measurements

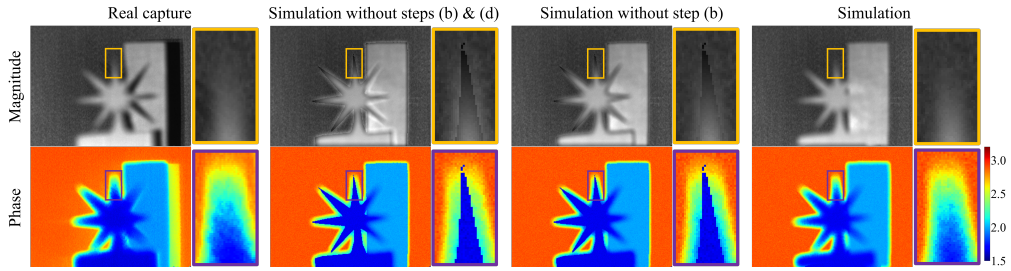
Defocus blurring occurs because a lens-sensor pair can only perfectly focus at a single depth plane in the scene called the *focal plane*. It is well-known that the incident defocused illumination from a given scene depth can be modeled by convolving the scene with a depth-dependent blur PSF (see, for example, chapter 7.4.4 of [35] and [36]).

A ToF camera computes depth maps by performing a series of operations, some nonlinear, on multiple captured measurements of the illumination incident on the camera. Because of these nonlinear operations, even if the blur on the incident illumination is linear, the blur on the computed depth map is ultimately not a linear operation. However, it can be shown that ToF defocus blur can be modeled as convolution performed in the phasor (or complex) domain of the ToF measurements. We provide the derivation of this defocused ToF forward model next. More details on the operation of a ToF system can be found in references such as [37, 38].

We first consider a ToF system that does not suffer from defocus blurring. It operates by first sending out a periodic illumination $I(t)$ to the scene, where t denotes time. For simplicity, suppose $I(t) = \cos(2\pi f t)$ for some frequency f. For an object at a distance d from the camera, the illumination travels a total distance of $2d$ as it is reflected by the object and back into the ToF sensor, which thus receives time-delayed illumination of the form $E(t) = I(t - \tau) = A\cos(2\pi f(t - \tau)) = A\cos(2\pi f t - \phi)$. Here, the amplitude A accounts for



(a) Simulation procedure



(b) Ablation simulations

Fig. 2. Top: Procedure for simulating large-aperture ToF measurements. The measurement is first converted to a phasor $\mathbf{X} = \mathbf{I} - i\mathbf{Q}$ and split into its depth components. Occluded regions are then interpolated by nearest-neighbor. The interpolated phasor and occlusion mask are then convolved by depth-dependent blur PSFs $\mathbf{P}_d$. Transparency masks obtained from the occlusion masks are then elementwise multiplied, and the outcomes at each depth are summed. Bottom: The simulation obtained by this method matches a real capture of a similar scene. Excluding the partial occlusion simulation (steps (b) and (d)) produces inaccurate simulations.

light absorption by the object and the phase ϕ is related to the object's distance d via:

$$\phi = 2\pi f \tau = \frac{4\pi f d}{c}, \tag{1}$$

where c is the speed of light. We assume that $d \leq \frac{c}{2f}$ to avoid the periodic ambiguity in d. Extending this framework to a 3D scene with objects at different lateral locations (x, y) and different distances $\mathbf{D}(x, y)$, the 2D received illumination is then denoted $\mathbf{E}(t) = \mathbf{A}(x, y) \circ \cos(2\pi f t - \mathbf{\Phi}(x, y))$ with $\mathbf{\Phi}(x, y) = \frac{4\pi f \mathbf{D}(x,y)}{c}$ where $\circ$ denotes elementwise multiplication.

The ToF camera measures the amplitude $\mathbf{A}$ and the phase $\mathbf{\Phi}$ of the returned illumination by correlating it with multiple periodic reference signals of the form $R_k(t) = \cos\left(2\pi f t + \frac{2\pi k}{N}\right)$ for some positive integer N and for non-negative integers $k < N$. For each reference signal and for some exposure time T, the sensor ultimately measures:

$$\mathbf{U}_k(x, y) = \frac{1}{T}\int_0^T R_k(t)\mathbf{E}(t)(x, y)\,\mathrm{d}t \approx \frac{\mathbf{A}(x, y)}{2}\cos\left(\mathbf{\Phi}(x, y) + \frac{2\pi k}{N}\right). \tag{2}$$

These individual measurements $\mathbf{U}_k$ are typically referred to as *quads*. Let $N \geq 3$ (typical ToF

cameras use values of 3, 4, or 6 for N), and define the following quantities:

$$\mathbf{I} = \frac{4}{N}\sum_{k=0}^{N-1} \mathbf{U}_k \cos\left(\frac{2\pi k}{N}\right), \qquad \mathbf{Q} = \frac{4}{N}\sum_{k=0}^{N-1} \mathbf{U}_k \sin\left(\frac{2\pi k}{N}\right). \tag{3}$$

It can then be shown (see supplementary material) that the corresponding complex value (or phasor) has magnitude $\mathbf{A}$ and phase $\mathbf{\Phi}$:

$$\mathbf{X} = \mathbf{I} - i\mathbf{Q} = \mathbf{A} \circ \exp(i\mathbf{\Phi}). \tag{4}$$

Thus, the magnitude and phase of the reflected illumination can be obtained:

$$\mathbf{A}(x,y) = \sqrt{\mathbf{I}(x,y)^2 + \mathbf{Q}(x,y)^2}, \qquad \mathbf{\Phi}(x,y) = \arctan\left(\frac{-\mathbf{Q}(x,y)}{\mathbf{I}(x,y)}\right),$$

and given $\mathbf{\Phi}$, the distance of each pixel of the scene can be calculated using Eq. (1). Finally, the depth map can be obtained from the measured distances via a calibration procedure such as capturing a flat wall at multiple depths. In this paper, we use "distance" and "depth" interchangeably noting that they are equivalent to a calibration procedure. Note that $\mathbf{\Phi}$ (and thus the ToF depth map) is a nonlinear function of the measured intensities.

When subject to defocus blurring, the illumination $\mathbf{E}(t)$ incident on the sensor is convolved by the blur PSF corresponding to the reflecting object's depth. Separating the two-dimensional reflected illumination $\mathbf{E}(t)$ into multiple matrices $\mathbf{E}_d(t)$ where $\mathbf{E}_d(t)(x,y) = \mathbf{E}(t)(x,y)$ if the scene depth at pixel (x,y) is equal to d and 0 otherwise, we can represent the blurred reflected illumination $\tilde{\mathbf{E}}(t)$ as:

$$\tilde{\mathbf{E}}(t) = \sum_{d\in\text{depths}} \mathbf{P}_d \circledast \mathbf{E}_d(t),$$

where $\mathbf{P}_d$ is the blur kernel for depth d, and $\circledast$ denotes 2D convolution. Since the quads $\mathbf{U}_i$ are linear in the reflected illumination $\mathbf{E}$ (Eq. (2)), and since the $\mathbf{I}/\mathbf{Q}$ measurements are linear in the quads (Eq. (3)), we can obtain the defocused $\tilde{\mathbf{I}}/\tilde{\mathbf{Q}}$ measurements by splitting the $\mathbf{I}/\mathbf{Q}$ measurements in the same way into $\mathbf{I}_d/\mathbf{Q}_d$ and applying the defocus blur convolution to them:

$$\tilde{\mathbf{I}} = \sum_{d\in\text{depths}} \mathbf{P}_d \circledast \mathbf{I}_d, \qquad \tilde{\mathbf{Q}} = \sum_{d\in\text{depths}} \mathbf{P}_d \circledast \mathbf{Q}_d. \tag{5}$$

Note that the blurred phase measurement of the ToF system is then given by:

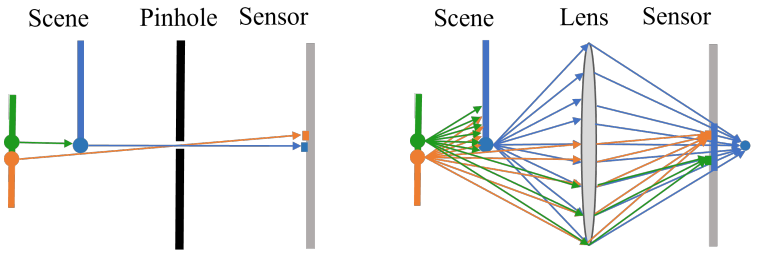
$$\tilde{\mathbf{\Phi}}(x,y) = \arctan\left(\frac{-\tilde{\mathbf{Q}}(x,y)}{\tilde{\mathbf{I}}(x,y)}\right) = \arctan\left(\frac{-\sum_{d\in\text{depths}} \mathbf{P}_d \circledast \mathbf{Q}_d}{\sum_{d\in\text{depths}} \mathbf{P}_d \circledast \mathbf{I}_d}\right).$$

Thus, although blurring is a linear operation on intensities, the resultant phase measurement $\tilde{\mathbf{\Phi}}$ is a nonlinear function of these blurred intensities. As such, blur affects ToF depth maps in a nonlinear fashion, a key point that differentiates ToF defocus deblurring from conventional image deblurring where defocus blur can be modeled as a linear function.

It is difficult to express $\tilde{\mathbf{\Phi}}$ as a direct function of $\mathbf{\Phi}$. Thus, instead of simulating the blurred phase (and depth) image directly, we leverage the phasor representation of ToF measurements. That is, denoting $\mathbf{X} = \mathbf{I} - i\mathbf{Q}$ and $\mathbf{X}_d = \mathbf{I}_d - i\mathbf{Q}_d$ and noting that $\mathbf{P}_d$ consists only of real numbers, the blurred measurement's phasor representation is:

$$\tilde{\mathbf{X}} = \tilde{\mathbf{I}} - i\tilde{\mathbf{Q}} = \sum_{d\in\text{depths}} \mathbf{P}_d \circledast \mathbf{X}_d. \tag{6}$$

Thus, given ground truth (all-in-focus) amplitude and depth maps, we can simulate a defocused ToF measurement by converting the ground truth pair into its phasor domain, splitting it into its depth components (step (a) in Fig. 2a), and applying Eq. (6) (step (c) in Fig. 2a).



(a) Pinhole diagram (b) Large aperture diagram

Fig. 3. Large-aperture partial occlusions. In a pinhole model (a), an object's illumination either fully reaches the sensor pixel (blue, orange) or is fully occluded (green). The resulting captured image contains all of the nonoccluded object's (orange) illumination but none of the occluded object's (green). For a large-aperture lens (b), objects can be partially occluded, where only part of the illumination from the object is occluded from the lens (orange, green). Partial occlusion can be modeled by multiplying partially occluded objects' total illumination by the fraction of nonocclusion.

3.2. Simulating large-aperture partial occlusions

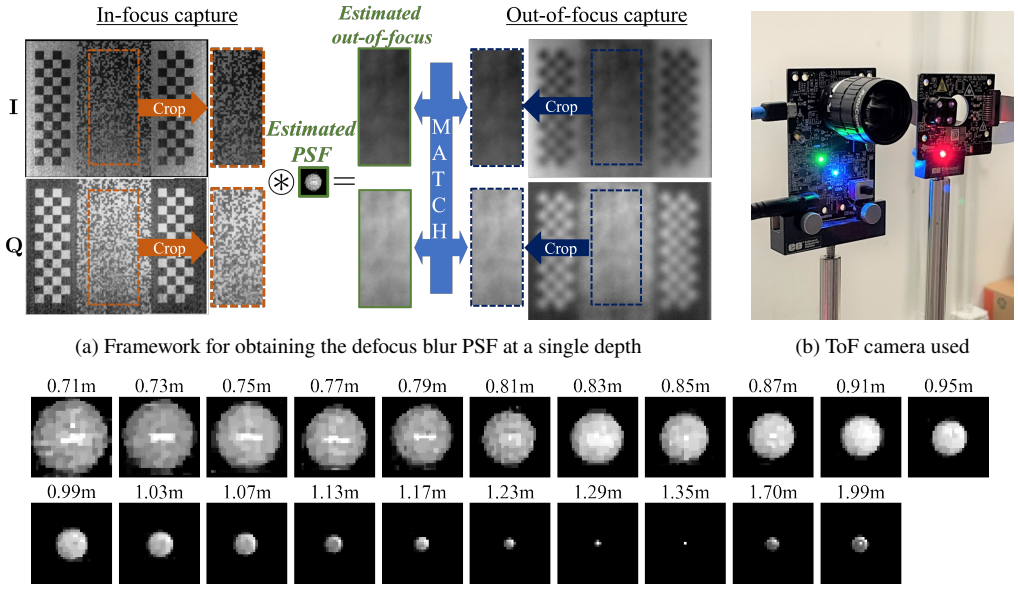
In addition to defocus blurring, another optical consequence of large apertures is partial occlusions, where a fraction of the illumination from occluded objects reaches the sensor [39], as illustrated in Fig. 3. Consider a point source P_1 at depth d_1 (e.g. green dot in Fig. 3) that gets projected to the sensor pixel (x_0, y_0). In a pinhole camera, there will be one ray from P_1 to (x_0, y_0). However, for a large aperture, multiple light rays from P_1 arrive at multiple points on the aperture that all then get refracted to (x_0, y_0). Consider adding another point source P_0 at depth $d_0 < d_1$ (e.g. blue dot in Fig. 3) that also gets projected to pixel (x_0, y_0). In a pinhole model, P_0 occludes the singular ray from P_1, and P_1 is completely occluded. However, for a large aperture, P_0 will only occlude a fraction of the light rays from P_1 to the aperture. The other light rays from P_1 still arrive at other portions of the aperture and get refracted onto (x_0, y_0). Thus, P_1 is only partially occluded, and a fraction of its illumination still reaches (x_0, y_0).

Suppose we have a ground truth all-in-focus pinhole image $\mathbf{E}$ of the incident light from a scene. In this image, all occluded pixels are completely occluded and thus absent in the image. We wish to simulate partial occlusion for such objects. There are two issues to address.

The first issue is that $\mathbf{E}$ does not contain the occluded values. In [40], the researchers propose estimating occluded values using interpolation techniques. We address the issue by using nearest neighbor interpolation to estimate the value of occluded pixels, as illustrated in step (b) of Fig. 2a. In particular, we split $\mathbf{E}$ into its depth components $\mathbf{E}_d$. For each depth d, we identify the occluded pixels: pixels for which there exists objects in $\mathbf{E}_{d'}$ for some $d' < d$. We then apply nearest neighbor interpolation to these pixels using $\mathbf{E}_d$ as the neighborhood.

The second issue is that the fraction of the occluded pixel that arrives at the sensor is unknown. For a partially occluded image $\mathbf{E}_d$ at depth d, we aim to find a transparency mask $\mathbf{T}_d$ where $\mathbf{T}_d(x, y)$ represents the fraction of illumination from $\mathbf{E}_d(x, y)$ incident at the lens aperture. The partially occluded image at depth d can then be modeled as $\tilde{\mathbf{E}}_d = \mathbf{T}_d \circ \mathbf{E}_d$, where $\circ$ denotes elementwise multiplication.

We approximate the transparency masks $\mathbf{T}_d$ for each depth d in the following way. We first generate an occlusion mask $\mathbf{O}_d$ at each depth d that is 1 at pixels where there is an object at the depth d and 0 everywhere else. We then convolve this occlusion mask by the blur PSFs for d to obtain the final occlusion mask: $\tilde{\mathbf{O}}_d = \mathbf{O}_d \circledast \mathbf{P}_d$ (step (c) of Fig. 2a). The transparency mask at depth d is then $\mathbf{T}_d = \max\left(0, 1 - \sum_{d'<d} \tilde{\mathbf{O}}_{d'}\right)$ (step (d) of Fig. 2a). That is, we partially occlude



(a) Framework for obtaining the defocus blur PSF at a single depth

(b) ToF camera used

(c) Calibrated PSFs

Fig. 4. Calibrating depth-dependent blur PSFs. (a) To obtain the lens's PSFs, pairs of in-focus and out-of-focus (focal plane of 1.4m) measurements of a random pattern (aligned using checkerboard corners) are captured. Eq. (8) is then solved to find the PSF that when convolved with the in-focus capture gives a result that is close to the out-of-focus capture. (b) The ToF camera used in this work is the Texas Instruments OPT8241-CDK-EVM with a 35mm $f/1.7$ lens. (c) The procedure is repeated with the pattern at multiple depths to obtain the shown PSFs. Each PSF image is 33×33 pixels.

an image at depth d by the occlusion masks of all the objects in front of it.

To apply the partial occlusion model to ToF measurements, we note that partial occlusion affects the incident illumination $\mathbf{E}(t)$. Since elementwise multiplication of the transparency mask is linear, the elementwise multiplication can be factored out of the phasor measurements (Eq. (4)). Thus, given all-in-focus phasor measurements $\mathbf{X}$, we obtain large-aperture ToF simulations with:

$$\tilde{\mathbf{X}} = \sum_{d \in \text{depths}} \mathbf{T}_d \circ (\mathbf{P}_d \circledast \mathbf{X}_d) \,. \tag{7}$$

Modeling partial occlusion is crucial to the simulator's effectiveness. In Fig. 2b, we show that excluding the simulation of partial occlusions results in simulations that do not match the real measurements of a physical large-aperture ToF system. In Section 5.1 and Fig. 6, we show that neural networks trained on simulations with partial occlusions greatly outperforms those trained on simulations without.

3.3. *Calibrating depth-dependent blur kernels*

To simulate defocus blurring and partial occlusions, one needs the defocus blur PSFs $\mathbf{P}_d$ at different depths d. We estimate the PSFs of our ToF camera using a calibration system based on ideas from [41,42] and illustrated in Fig. 4. We choose a set of depths $S = \{d_1, d_2, ..., d_K\}$ whose PSFs we wish to obtain. For each chosen depth $d_i \in S$, we capture a binary random pattern with checkerboard patterns on the edges with our ToF camera. This capture will be blurred by the blur PSF at d_i. We then focus the lens at d_i and capture a sharp version of the random pattern. For the calibration, we use the phasor representation of the measurements $\mathbf{X} = \mathbf{I} - i\mathbf{Q}$.

Given both the sharp and blurry phasor measurements of the calibration patterns at a depth d, we first align the measurements by matching corner points on the checkerboard patterns. Afterwards, the random pattern is cropped for both measurements (denoted $\mathbf{X}_{d,\text{sharp}}$ and $\mathbf{X}_{d,\text{blur}}$) and the blur kernel $\mathbf{P}_d$ is estimated by solving the following optimization:

$$\begin{aligned} \mathbf{P}_d = \arg\min_{\mathbf{P}} \|\mathbf{P} \circledast \mathbf{X}_{d,\text{sharp}} - \mathbf{X}_{d,\text{blur}}\|_2^2 + \lambda \text{TV}(\mathbf{P}), \\ \text{s.t. } \mathbf{P} \geq 0 \end{aligned} \tag{8}$$

where $\text{TV}(\mathbf{P}) = \|\nabla_x \mathbf{P}\|_1 + \|\nabla_y \mathbf{P}\|_1$ is the total variation, and λ is a hyperparameter. We set $\lambda = 10^{-6}$ in our experiments. The optimization is solved in an iterative fashion where for each iteration, a step is taken with the ADAM optimizer [43] for the objective function and then all negative values of $\mathbf{P}$ are set to 0. This procedure is repeated for all chosen depths in S to obtain the results in Fig. 4c.

We calibrate only for 21 depth values for computational efficiency even if scenes can contain more, in fact, continuous, depth values. Thus, in simulating the large-aperture ToF measurements in Eq. (7), we split the ground truth phasor measurements into 21 $\mathbf{X}_d$ components according to which $d \in S$ each pixel's depth is closest to. We use only these 21 PSFs as is and do not perform any PSF interpolation for depths in between.

4. Deblurring neural network

4.1. Network architecture

EDoF-ToF operates by inputting the large-aperture ToF measurements into a deblurring network to output all-in-focus magnitude and depth maps. There are multiple options for the deblurring neural network architecture. In this work, we use the scale-recurrent network (SRN) architecture proposed by [33], an architecture shown to have excellent performance on conventional image deblurring. We train the SRN from scratch using the simulated data described in Section 3.

As input, we feed in data with two channels: the $\mathbf{I}$ and $\mathbf{Q}$ values of the ToF measurement as described in section 3.1. These are simply the real and (negative) imaginary parts of the phasor representation of the ToF measurement (Eq. (4)). We train two SRNs: one to output the magnitude image $\mathbf{A}$ and one to output the depth map $\mathbf{D}$.

The SRN is a multi-scale network, so it takes in the $\mathbf{I}/\mathbf{Q}$ inputs at multiple scales, and for each scale, outputs the magnitude or depth maps for those same scales. At the very first scale, the $\mathbf{I}/\mathbf{Q}$ is downscaled to $\frac{1}{4}$ its original size and input into the network to obtain the magnitude or depth map estimation at $\frac{1}{4}$ scale. The estimate is then upsampled to $\frac{1}{2}$ scale by bilinear interpolation and appended to the original $\mathbf{I}/\mathbf{Q}$ measurements downscaled by $\frac{1}{2}$. Now, this 3-channel input ($\mathbf{I}$, $\mathbf{Q}$, magnitude or depth) is input to the network. The output estimate is again upsampled, appended to the original $\mathbf{I}/\mathbf{Q}$ measurements, and passed into the SRN to give the final original-scale estimate.

We use the same original hyperparameters proposed by [33] for the SRN architecture. The SRN architecture contains 3 components: an encoder, a convolutional long-short term memory (LSTM) cell [44], and a decoder. The encoder is composed of three "EBlocks", which each consist of a convolutional layer and 3 "ResBlocks". The first convolutional layer has a stride of 1 for the first EBlock and 2 for the next two EBlocks. Each ResBlock consists of 2 stride-1 convolutional layers that are trained on the residuals of the inputs. The EBlocks output 32, 64 and 128 channels, respectively. The decoder consists of two DBlocks followed by an output block. The Dblocks each consist of two ResBlocks followed by a stride-2 deconvolution layer and output 64 and 32 channels, respectively. The output block is similar to a DBlock with the deconvolution layer replaced by a stride-1 convolutional layer that provides the final output. All convolutions have kernel sizes of 5×5 and ReLU activations. Skip connections connect EBlocks and DBlocks with the corresponding data sizes. The same network weights are used for all three scales.

4.2. Training details

For training data, we simulate large-aperture ToF measurements (detailed in Section 3) on the FlyingThings3D dataset, which is a synthetic dataset containing more than 21,000 data points with both scene intensity (RGB) and depth [45]. For each iteration, we perform a random crop of the scene and resize it to (240, 320) and convert the intensity to grayscale. The scene crop's depth is then normalized and biased to have depth values in $[0.65\text{m}, 18\text{m}]$ when training the magnitude SRN and $[0.65\text{m}, 2\text{m}]$ when training the depth SRN. We then divide the intensity at each pixel by the squared depth at that pixel to account for the $\frac{1}{R^2}$ intensity fall-off of the ToF active illumination. Next, we convert the depth measurements into phase measurements using Eq. (1) and then obtain the phasor measurements $\mathbf{X}$ using Eq. (4). Random Gaussian noise is added to the phasor measurements, which are then normalized to unit magnitude. Finally, the simulated large-aperture ToF measurements are obtained by applying Eq. (7) to $\mathbf{X}$.

The magnitude SRN is trained with a sum of pixel-wise ℓ_1 loss and perceptual loss, which is taken to be the mean-squared error loss for the second-layer and fourth-layer outputs of the VGG16 network [46]. The perceptual loss is weighed by 0.1. The depth SRN is trained only with pixel-wise ℓ_1 loss as we find that including perceptual loss makes no significant difference on the output depth map estimates.

Training is done using Pytorch [47] and the AdamW optimizer [48] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for 80 epochs, where each epoch is a full pass through the FlyingThings3D training dataset. The learning rate is initially set to 5×10^{-5} and then decreased to 10^{-5} after 40 epochs and then to 2×10^{-6} after another 25 epochs.

5. Experiments

We evaluate EDoF-ToF on a Texas Instruments OPT8241-CDK-EVM ToF camera equipped with an Arducam 35mm $f/1.7$ lens focused at 1.4m. We use a single modulation frequency of $f = 48$MHz. Aside from the experiment in Section 5.3, all measurements and results shown use one capture at a frame rate of 10-15fps. We use the term "conventional" to refer to the OPT8241 camera's measurement without the neural network enhancement.

"Ground truth" images displayed in subsequent figures are obtained by focusing the lens at the object of interest. We note that this is not a perfect ground truth since these measurements still have narrow DoFs, leading to blurred backgrounds and flying pixels around the foreground object. Nonetheless, these comparison images show the detail of the foreground objects EDoF-ToF is able to digitally reconstruct.

5.1. Ablation experiments

Network architecture. First, we experiment with different network architectures for the deblurring neural network. The first architecture we experiment is the u-net architecture [49], a common architecture used in image-to-image tasks. The u-net we employ contains 5 downsampling blocks (outputting 128, 128, 256, 512, and 1024 channels, respectively) followed by five upsampling blocks (outputting 512, 256, 128, 128, and 1 channels, respectively). Each block contains two convolutional layers with ReLU activations. Skip connections connect the downsampling layers with the upsampling layers that have the corresponding input size. We use separate u-nets for reconstructing the magnitude and depth maps. The second architecture we experiment is a joint SRN: one SRN network that outputs two channels pertaining to the magnitude and depth maps. Finally, we test the separate SRNs architecture where we have separate SRNs for magnitude and depth. All networks are trained with the same training parameters (loss function, learning rate, number of epochs, etc.).

This comparison, presented in Fig. 5, shows how all three architectures are able to significantly deblur the input. Visually, we find that separate SRNs for magnitude and depth output the least

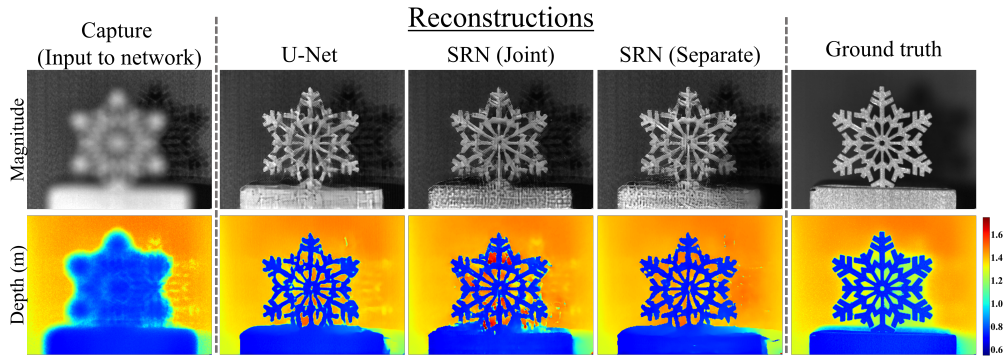


Fig. 5. Reconstructions with different network architectures: the u-net [49], a single joint SRN [33] that simultaneously outputs both magnitude and depth, and two SRNs that separately output magnitude and depth. All architectures are able to perform significant deblurring, with the separate SRNs giving the least noisy reconstructions.

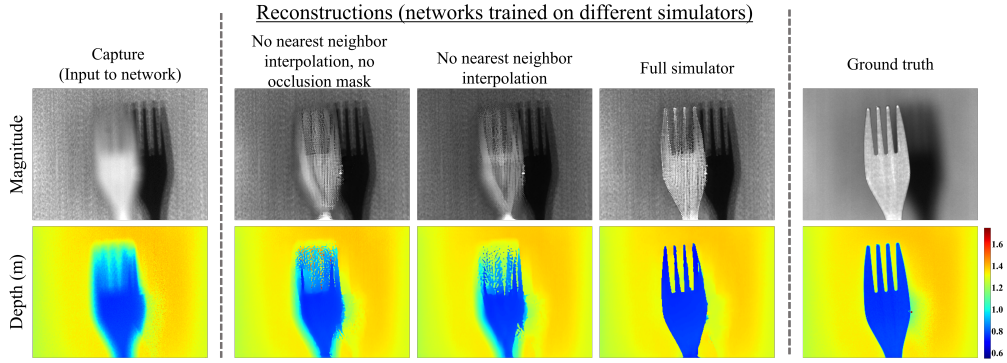


Fig. 6. Reconstructions of SRNs trained without and with partial occlusion simulation. Removing the nearest neighbor interpolation or the occlusion mask used to simulate partial occlusions degrades reconstructions, highlighting the importance of modeling partial occlusions when training the networks. The fork's depth is approximately 0.71m.

noisy reconstructions. However, it is clear that the success of EDoF-ToF is not heavily dependent on the SRN's structure, which can be replaced with other deblurring network architectures, especially new ones as they arise. We use separate SRNs for all subsequent experiments.

Effect of modeling partial occlusions. Next, we perform an ablation experiment that highlights the importance of modeling partial occlusions in our training data simulator. In Section 3.2 and Fig. 2b, we discuss how modeling partial occlusions in our simulator gives simulations that more accurately match real captures. Our modeling of partial occlusions consists of two steps: (i) nearest neighbor interpolation to estimate values of occluded pixels and (ii) applying an occlusion mask convolved with the PSFs to model the extent of occlusion. These are steps (b) and (d) in Fig. 2a.

For this experiment, we train SRNs without performing the nearest neighbor interpolation (step b in Fig. 2) and SRNs without performing both the nearest neighbor interpolation and the application of occlusion masks (steps b and d in Fig. 2). We show in Fig. 6 that the reconstructions trained by such simulations are of significantly lower accuracy. This shows the importance of modeling partial occlusions and how each step of our simulator is crucial for accurate training of the neural networks.

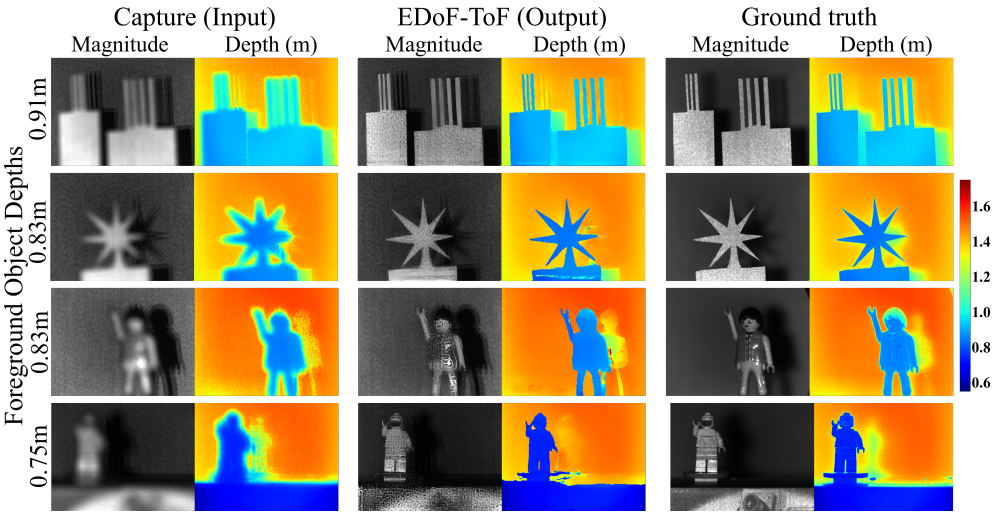


Fig. 7. Reconstructions of out-of-focus objects at different depths by EDoF-ToF. Left: Measurement captured by the ToF camera focused at 1.4m (input to the neural network). Middle: EDoF-ToF reconstructions. Right: Measurement captured by the ToF camera focused at the foreground object for comparison. EDoF-ToF is able to provide sharp and accurate depth maps by significantly deblurring objects outside of the lens's DoF and reconstructing fine details such as the hand poses on the bottom two toys.

5.2. Experiments for various real scenes

Objects at a single defocused depth. Next, we capture various scenes with real objects and evaluate the quality of our EDoF-ToF's reconstructions. We first capture individual objects outside the camera's focal range and pass the measurements to the trained SRNs. We compare them to conventional ToF measurements with the lens focused on the object. The results, shown in Fig. 7, together with the results in Figs. 5 and 6, demonstrate that EDoF-ToF produces sharp depth maps, deblurring even objects that are around 0.75m from the camera, which our calibration in Fig. 4c shows has a blur diameter of around 30 pixels. EDoF-ToF also solves the issue of flying pixels: the halo of incorrect depth estimates around the foreground object observed in the in-focus capture. We note that at large blurs, the reconstructed magnitude maps may have artifacts. However, these may be acceptable for many applications since ToF cameras are primarily used to obtain depth maps.

Multiple objects at multiple depths. In our next experiment, we capture scenes with multiple objects at different depths. The results, displayed in Fig. 8, show that EDoF-ToF can take in measurements containing objects with different degrees of focus/blur and output an all-in-focus reconstruction.

Comparison to other methods. In Fig. 9, we compare EDoF-ToF with two alternative methods. The first method is to apply an off-the-shelf conventional image deblurring method directly on the captured magnitude and depth maps. We use the pre-trained SRN from [33], a network trained on a dataset of averaged images from videos. The failure of using this off-the-shelf pre-trained deblurring method shows how it is important to train the deblurring network on the calibrated blurs of the ToF camera in hand. The second method is that of Xiao et al. [20], who in their work perform deblurring of large-aperture ToF measurements using an optimization framework based on ADMM. Their work handles smaller blurs than the ones we have demonstrated for our method, and indeed, we find that while their method can reduce small blurs (such as that of the hand and grating), it is unable to accurately reconstruct objects with

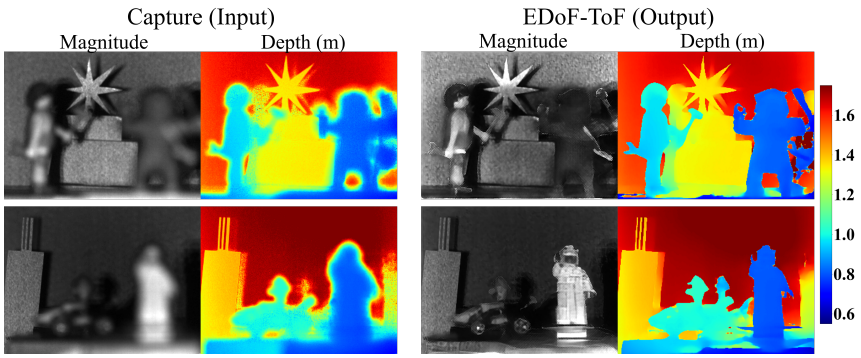


Fig. 8. Captures and reconstructions of scenes with multiple objects at different depths. EDoF-ToF is able to simultaneously reconstruct objects blurred by different PSFs. Fine details, such as the tools the man is holding (top) and the hand poses of the bear (top) and the man (bottom), are clearly revealed.

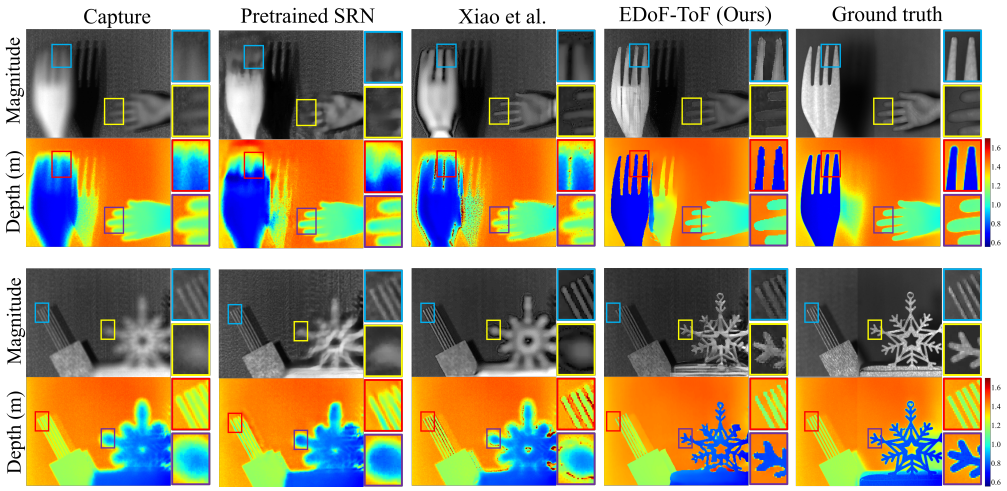
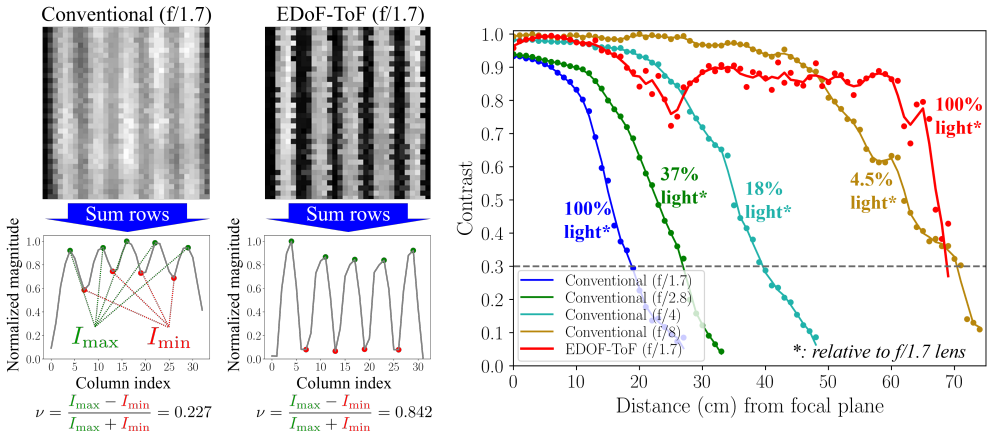


Fig. 9. Comparisons of EDoF-ToF to using the pre-trained SRN from [33] and to the methd of Xiao et al. [20]. EDoF-ToF can handle larger defocus blurs than both methods. The ground truth images are obtained by manually stitching two measurements each focused at one of the two objects and are presented for comparison.

large blurs (such as that of the fork and snowflake) that our method is able to handle.

5.3. Quantifying depth of field

Finally, we quantify the DoF improvement of EDoF-ToF over a conventional ToF system. To quantify the DoF of a given system, we capture 3-pixel wide wooden gratings with 3-pixel gaps in between (6 pixels per line pair) at various depths. For each capture, we sum along the length of the gratings to obtain a 1D average, as shown in Fig. 10a. We then average the peaks of this 1D signal to acquire $I_{\max}$ and the troughs to acquire $I_{\min}$. Essentially, $I_{\max}$ refers to the average grating measurement value, and $I_{\min}$ refers to the average gap measurement value. We then compute the contrast as $\nu = \frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}}$. The DoF of a system is the depth range in front of the focal plane of 1.4m such that the contrast ν of the measurement of the grating placed in that range is at least 0.3. This metric is analogous to the MTF30 metric typically used to characterize the



(a) Contrast calculation procedure

(b) Contrast plots for EDoF-ToF and a conventional ToF system

Fig. 10. Experiment to quantify DoF. (a) Sample of a conventional $f/1.7$ ToF system's and EDoF-ToF's magnitude measurements of 3-pixel wide gratings 20cm in front of the focal depth. The measurements are summed row-wise, and the peaks and troughs are averaged to obtain I_{min} and I_{max} that are then used to calculate the contrasts ν. (b) Contrast plot for EDoF-ToF and a conventional ToF system with different apertures. The contrasts are experimentally measured by capturing gratings placed at various depths (in intervals of 1cm) in front of the focal plane of 140cm. The DoF is the depth range at which $\nu \geq 0.3$ and is plotted in Fig. 1. The light throughput shown for each system is taken as the ratio of its lens aperture area to that of a $f/1.7$ lens. EDoF-ToF has a 3.6× larger DoF than a conventional ToF system with the same lens and has a DoF on par with a conventional $f/8$ ToF system, which has 4.5% the light throughput.

resolution of an imaging system.

We perform the experiment for our EDoF-ToF system equipped with an $f/1.7$ lens, as well as for the conventional OPT8241 camera with different lens apertures. As shown in Fig. 10b, EDoF-ToF achieves a DoF of 69cm, which is around 3.6× larger than that of a conventional system with the same $f/1.7$ lens (19cm). EDoF-ToF with a $f/1.7$ lens essentially achieves a DoF that is on par with a conventional ToF system equipped with an $f/8$ lens. Since light throughput increases by the square of the aperture diameter and the f-number is inversely proportional to the aperture diameter, the EDoF-ToF system thus receives $\frac{8^2}{1.7^2} \approx 22.1\times$ more light than the conventional $f/8$ ToF system with a similar DoF. The DoF and relative light throughput of these different systems are plotted in Fig. 1.

6. Discussion and conclusion

Limitations. One may observe artifacts in both conventional ToF measurements and EDoF-ToF reconstructions in shadowed regions where the ToF system's active illumination does not reach due to being blocked by other objects. Since almost no photons are reflected from these areas, phase information is not sufficiently measured to calculate depth values. However, this ToF shadow problem can be addressed using techniques such as filtering, filling, or inpainting [50–52], and EDoF-ToF can also be extended using similar techniques to address shadowed regions.

The simulation mathematical model presented in Section 3 assumes that all objects in the scene are diffuse such that they reflect light equally in all directions. Objects that are purely reflective essentially only reflect one ray of light back and would neither experience defocus blur (as the single ray remains focused) nor partial occlusions (since the single ray only hits one

point on the aperture). The model also does not handle transparent objects that refract ToF's illumination to other parts of the scene and increase the apparent depth. Handling reflective and transparent objects are active problems for conventional ToF [53, 54] and have not been explored in our EDoF-ToF.

While we argue that a larger aperture with its higher light throughput can allow lower on-device power consumption while maintaining high SNR by decreasing the amount of emitted light, the neural network computation involved in EDoF-ToF also adds required power consumption. This study does not perform a detailed power analysis and does not account for the power requirements of the deep network. We note, however, that the SRN computations can be performed offline on a server remote from the device, potentially removing the requirement of increased power consumption by the camera device itself.

Conclusion. This work shows that by combining deep learning methods with a large-aperture lens, EDoF-ToF can achieve high light-throughput and wide depth-of-field 3D CWAM ToF imaging. Such a feature allows increased SNR for a wide depth of field, potentially enabling lower power consumption (by allowing less light to be emitted), increased frame rate (by allowing smaller exposure times), and overall increased measurement accuracy. The key idea is to train a deblurring neural network using realistic large-aperture ToF simulations that account for the lens's blur and partial occlusions. The depth cues from ToF measurements assist the deblurring network, without needing any additional optics.

EDoF-ToF continues in the line of work enhancing the capabilities of ToF cameras using deep learning. Other works in literature have shown that deep learning can also solve other tasks such as denoising, multi-path interference correction, and phase unwrapping. These examples show that even with minimal changes to hardware, significant enhancements can be achieved with algorithmic processing on ToF cameras. Indeed by integrating both algorithmic and hardware innovations, 3D imaging continues to reach new heights to provide richer representations of the world around us for use in numerous applications.

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

See Supplement 1 for supporting content.

References

1. R. Lange and P. Seitz, "Solid-state time-of-flight range camera," IEEE J. Quantum Electron. **37**, 390–397 (2001).
2. J. K. Aggarwal and L. Xia, "Human activity recognition from 3d data: A review," Pattern Recognit. Lett. **48**, 70–80 (2014).
3. U. B. Himmelsbach, T. M. Wendt, and M. Lai, "Towards safe speed and separation monitoring in human-robot collaboration with 3d-time-of-flight cameras," in *IEEE Int. Conf. Robot. Comput.*, (IEEE, 2018), pp. 197–200.
4. J. Penne, C. Schaller, J. Hornegger, and T. Kuwert, "Robust real-time 3d respiratory motion detection using time-of-flight cameras," Int. J. Comput. Assist. Radiol. Surg. **3**, 427–431 (2008).
5. L. Jia and R. J. Radke, "Using time-of-flight measurements for privacy-preserving tracking in a smart room," IEEE Trans. Ind. Informat. **10**, 689–696 (2013).
6. A. Sioma, "3D imaging methods in quality inspection systems," in *Photon. Appl. Astronomy, Commun., Ind., High-Energ. Phys. Experiments,* vol. 11176 R. S. Romaniuk and M. Linczuk, eds., International Society for Optics and Photonics (SPIE, 2019), pp. 150 – 159.
7. S. Zhou and S. Xiao, "3d face recognition: a survey," Human-centric Comput. Inf. Sci. **8**, 1–27 (2018).
8. G. Häusler, "A method to increase the depth of focus by two step image processing," Opt. Commun. **6**, 38–42 (1972).
9. E. R. Dowski and W. T. Cathey, "Extended depth of field through wave-front coding," Appl. Opt. **34**, 1859–1866 (1995).
10. S. Kuthirummal, H. Nagahara, C. Zhou, and S. K. Nayar, "Flexible depth of field photography," IEEE Trans. Pattern Anal. Mach. Intell. **33**, 58–71 (2010).
11. T. Instruments, *Introduction to the Time-of-Flight (ToF) System Design User's Guide*, Texas Instruments Incorporated.
12. R. E. Jacobson, S. F. Ray, G. G. Attridge, and N. R. Axford, *The Manual of Photography* (Focal Press, Oxford, 2000).

13. S. Tan, Y. Wu, S.-I. Yu, and A. Veeraraghavan, "Codedstereo: Learned phase masks for large depth-of-field stereo," arXiv preprint arXiv:2104.04641 (2021).
14. L. Jin, Y. Tang, Y. Wu, J. B. Coole, M. T. Tan, X. Zhao, H. Badaoui, J. T. Robinson, M. D. Williams, A. M. Gillenwater, R. R. Richards-Kortum, and A. Veeraraghavan, "Deep learning extended depth-of-field microscope for fast and slide-free histology," Proc. Nat. Acad. Sci. **117**, 33051–33060 (2020).
15. O. Cossairt and S. Nayar, "Spectral focal sweep: Extended depth of field from chromatic aberrations," in *IEEE Int. Conf. Comput. Photography,* (IEEE, 2010), pp. 1–8.
16. O. Cossairt, C. Zhou, and S. Nayar, "Diffusion coded photography for extended depth of field," ACM Trans. Graph. **29** (2010).
17. V. Sitzmann, S. Diamond, Y. Peng, X. Dun, S. Boyd, W. Heidrich, F. Heide, and G. Wetzstein, "End-to-end optimization of optics and image processing for achromatic extended depth of field and super-resolution imaging," ACM Trans. Graph. **37**, 1–13 (2018).
18. S. Jayasuriya, A. Pediredla, S. Sivaramakrishnan, A. Molnar, and A. Veeraraghavan, "Depth fields: Extending light field techniques to time-of-flight imaging," in *Int. Conf. 3D Vision,* (IEEE, 2015), pp. 1–9.
19. S. Honnungar, J. Holloway, A. K. Pediredla, A. Veeraraghavan, and K. Mitra, "Focal-sweep for large aperture time-of-flight cameras," in *Int. Conf. Image Process.*, (IEEE, 2016), pp. 953–957.
20. L. Xiao, F. Heide, M. O'Toole, A. Kolb, M. B. Hullin, K. Kutulakos, and W. Heidrich, "Defocus deblurring and superresolution for time-of-flight depth cameras," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, (2015), pp. 2376–2384.
21. J. Marco, Q. Hernandez, A. Munoz, Y. Dong, A. Jarabo, M. H. Kim, X. Tong, and D. Gutierrez, "Deeptof: off-the-shelf real-time correction of multipath interference in time-of-flight imaging," ACM Trans. Graph. **36**, 1–12 (2017).
22. S. Su, F. Heide, G. Wetzstein, and W. Heidrich, "Deep end-to-end time-of-flight imaging," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, (2018), pp. 6383–6392.
23. G. Agresti, H. Schaefer, P. Sartor, and P. Zanuttigh, "Unsupervised domain adaptation for tof data denoising with adversarial learning," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, (2019), pp. 5584–5593.
24. G. Agresti and P. Zanuttigh, "Deep learning for multi-path error removal in tof sensors," in *Proc. European Conf. Comput. Vision Workshops,* L. Leal-Taixé and S. Roth, eds. (Springer International Publishing, Cham, 2019), pp. 410–426.
25. F. Gutierrez-Barragan, H. Chen, M. Gupta, A. Velten, and J. Gu, "itof2dtof: A robust and flexible representation for data-driven time-of-flight imaging," arXiv preprint arXiv:2103.07087 (2021).
26. Q. Guo, I. Frosio, O. Gallo, T. Zickler, and J. Kautz, "Tackling 3d tof artifacts through learning and the flat dataset," in *Proc. European Conf. Comput. Vision,* (2018), pp. 368–383.
27. M. Poggi, G. Agresti, F. Tosi, P. Zanuttigh, and S. Mattoccia, "Confidence estimation for tof and stereo sensors and its application to depth data fusion," IEEE Sensors J. **20**, 1411–1421 (2019).
28. S. Song and H. Shim, "Depth reconstruction of translucent objects from a single time-of-flight camera using deep residual networks," in *Asian Conf. Comput. Vision,* (Springer, 2018), pp. 641–657.
29. I. Chugunov, S.-H. Baek, Q. Fu, W. Heidrich, and F. Heide, "Mask-tof: Learning microlens masks for flying pixel correction in time-of-flight imaging," arXiv preprint arXiv:2103.16693 (2021).
30. D. Qiu, J. Pang, W. Sun, and C. Yang, "Deep end-to-end alignment and refinement for time-of-flight rgb-d module," in *Proc. IEEE Int. Conf. Comput. Vision,* (2019), pp. 9994–10003.
31. J. Koh, J. Lee, and S. Yoon, "Single-image deblurring with neural networks: A comparative survey," Comput. Vis. Image Underst. **203**, 103134 (2021).
32. S. Nah, T. Hyun Kim, and K. Mu Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, (2017), pp. 3883–3891.
33. X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, "Scale-recurrent network for deep image deblurring," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, (2018), pp. 8174–8182.
34. O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better," in *Proc. IEEE/CVF Int. Conf. Comput. Vision,* (2019), pp. 8878–8887.
35. J. W. Goodman, *Introduction to Fourier optics* (W. H. Freeman and Company, 2017).
36. M. Potmesil and I. Chakravarty, "A lens and aperture camera model for synthetic image generation," ACM SIGGRAPH Comput. Graph. **15**, 297–305 (1981).
37. L. Li, "Time-of-flight camera–an introduction," Tech. white paper (2014).
38. M. Hansard, S. Lee, O. Choi, and R. P. Horaud, *Time-of-flight cameras: principles, methods and applications* (Springer Science & Business Media, 2012).
39. R. L. Cook, T. Porter, and L. Carpenter, "Distributed ray tracing," in *Proc. Annu. Conf. Comput. Graphs Interactive Tech.*, (1984), pp. 137–145.
40. B. A. Barsky, M. J. Tobias, D. R. Horn, and D. P. Chu, "Investigating occlusion and discretization problems in image space blurring techniques," in *Int. Conf. Vision, Video, Graphics,* (Citeseer, 2003), pp. 97–102.
41. Y. Wu, V. Boominathan, H. Chen, A. Sankaranarayanan, and A. Veeraraghavan, "Phasecam3d—learning phase masks for passive single view depth estimation," in *IEEE Int. Conf. Comput. Photography,* (IEEE, 2019), pp. 1–12.
42. F. Heide, M. Rouf, M. B. Hullin, B. Labitzke, W. Heidrich, and A. Kolb, "High-quality computational imaging through simple lenses," ACM Trans. Graph. **32**, 1–14 (2013).

43. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980 (2014).
44. S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Adv. Neural Inf. Process. Syst.*, (2015), pp. 802–810.
45. N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *IEEE Int. Conf. Comput. Vision Pattern Recognit.*, (2016). ArXiv:1512.02134.
46. J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conf. Comput. Vision,* (Springer, 2016), pp. 694–711.
47. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," arXiv preprint arXiv:1912.01703 (2019).
48. I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Int. Conf. Learning Representations,* (2019).
49. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Int. Conf. Med. Imag. Comput. Comput.-assisted Intervention,* (Springer, 2015), pp. 234–241.
50. Y. Yu, Y. Song, Y. Zhang, and S. Wen, "A shadow repair approach for kinect depth maps," in *Asian Conf. Comput. Vision,* (Springer, 2012), pp. 615–626.
51. W. Kazmi, S. Foix, and G. Alenya, "Plant leaf imaging using time of flight camera under sunlight, shadow and room conditions," in *Proc. IEEE Int. Symp. Robot. Sens. Environments,* (IEEE, 2012), pp. 192–197.
52. D. Zhang, Y. Yao, D. Zang, and Y. Chen, "A spatio-temporal inpainting method for kinect depth video," in *IEEE Int. Conf. Signal Image Process. Appl.*, (IEEE, 2013), pp. 67–70.
53. K. Kim and H. Shim, "Robust approach to reconstructing transparent objects using a time-of-flight depth camera," Opt. express **25**, 2666–2676 (2017).
54. K. Tanaka, Y. Mukaigawa, H. Kubo, Y. Matsushita, and Y. Yagi, "Recovering transparent shape from time-of-flight distortion," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition,* (2016), pp. 4387–4395.